



Centro Universitario de Ciencias Exactas e Ingenierías.

Ciencias de la computación

Seminario de Uso, Adaptación y Explotación de Sistemas Operativos, Becerra
Velázquez Violeta del Rocío

Docente: Becerra Velázquez Violeta del Rocío

Hernandez Lomelí Diego Armando

Estrada Ceja Ulises Iván

219750396

217230905

INNI- Ingeniería en informática

D02

Actividad de aprendizaje 8

Conociendo los procesos, hilos y su entorno

2023/09/30

Tabla de contenido

Introducción	3
1- Definición de procesos y su jerarquía	4
2- Estados de un proceso	5
a) Diagrama de estados de un proceso:	5
b) Definición de cada estado y sus transiciones válidas:	5
3- Descripción de los procesos y sus elementos:	6
4. Control de procesos	8
5- Definición de hilo.	10
6- Hilo vs. Proceso.	10
7- Gestión de hilos.	10
Conclusiones	13
Bibliografía	14
Anexos:	14

Tabla de imágenes:

Ilustración 1.- Diagrama de estados de un proceso genérico.	5
Ilustración 2.- Transición entre ejecución de un proceso a otro.	11

Introducción

Un proceso se define como un programa en ejecución compuesto de una cantidad de elementos, además de ser una entidad activa que usa recursos asignados por el sistema operativo y que puede encontrarse bajo distintos estados durante su ejecución. Su jerarquía se ve reflejada en los procesos que generan otros procesos, y éstos a su vez generan otros más; al que genera se le llama proceso padre y el generado es el proceso hijo. Los procesos pasan por varios estados y experimentan transiciones al pasar de un estado a otro dependiendo de factores como el uso de memoria, el procesador, eventos de entrada y salida, etc. En cualquier instante del tiempo de ejecución de un proceso, éste se identifica por una serie de elementos que dependen de datos en el proceso, de la memoria, de la CPU, de datos en la CPU, entre otros factores.

El SO debe hacerse cargo de los procesos mediante la asignación de distintos modos de ejecución, ya que debe protegerse a sí mismo del uso de instrucciones en un contexto inadecuado.

Los hilos se definen como un flujo de ejecución de tareas que se ejecutan secuencialmente, se relacionan con los procesos en que éstos últimos son una colección de hilos, ya que con esto podemos manejar mejor la modularidad de un sistema.

1- Definición de procesos y su jerarquía

Un proceso es un programa en ejecución que comparte e intercambia información mediante recursos asignados por el sistema operativo. Cada proceso se encuentra, en un instante dado, en uno de los tres distintos estados de ejecución: Listo, Ejecutando y Bloqueado.

En muchos sistemas operativos, el concepto tradicional de proceso se ha dividido en dos partes:

1. La propiedad de los recursos (proceso).
2. La ejecución del flujo de instrucciones (hilo o thread).

Se han sugerido varias definiciones para el término proceso:

- Un programa en ejecución.
- Una instancia de un programa ejecutado en un ordenador.
- La entidad asignable y ejecutable en un procesador.
- Una unidad de actividad, caracterizada por ejecutar una serie de instrucciones, tener un estado actual y tener asociados un conjunto de recursos del sistema.

También se puede ver a un proceso como una entidad que consiste en un número de elementos, de los cuales dos serían los elementos esenciales: el código de programa (que puede ser compartido entre procesos que ejecuten el mismo programa) y un conjunto de datos asociados a dicho código. Un proceso, a diferencia de un programa ejecutable, es una entidad activa, con un contador de programa que especifica la próxima instrucción a ejecutar y un grupo de recursos asociados.

Se puede decir que un proceso se compone de código de programa y datos asociados, además del Bloque de control del proceso (BCP, formado por una lista de elementos que se explican más adelante) que el SO crea y gestiona.

En cuanto a su jerarquía, es habitual que un proceso cree muchos otros procesos conforme se ejecuta. Por ejemplo, un proceso de aplicación puede generar otro proceso para recibir los datos que la aplicación está generando y para organizar dichos datos apropiadamente para su posterior análisis. El nuevo proceso se ejecuta en paralelo con el proceso original y se activa cuando los nuevos datos se encuentran disponibles. Cuando un SO crea un proceso a petición explícita de otro proceso, se le llama *creación del proceso*.

Cuando un proceso lanza otro, al primero se le llama *proceso padre*, y al segundo se le denomina *proceso hijo*. Para lograr la relación entre ambos, se requiere comunicación y cooperación entre ellos. Cada proceso tiene un identificador único denominado Process ID (PID), el cual distingue al proceso del resto de procesos del sistema. Los procesos pueden compartir recursos como ficheros y dispositivos E/S, y se comunican unos con otros mediante mecanismos

de comunicación interproceso que les da el SO. El proceso padre tiene un determinado nivel de control sobre sus hijos, ya que puede mandarles datos para ser procesados por ellos, terminar los procesos hijos o esperar a que algunos de ellos terminen para luego continuar con la propia ejecución del proceso padre.

2- Estados de un proceso

Conforme se ejecuta un proceso, éste va cambiando de estado. Su estado se define, en parte, por la actividad actual de dicho proceso. Cada proceso puede estar en uno de los siguientes estados:

a) Diagrama de estados de un proceso:

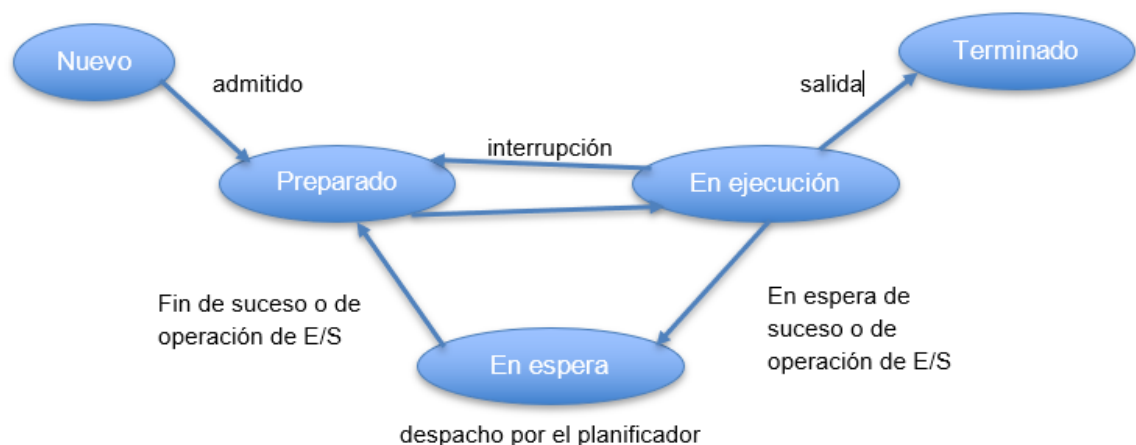


Ilustración 1.- Diagrama de estados de un proceso genérico.

b) Definición de cada estado y sus transiciones válidas:

Nuevo: El proceso se está creando y aún no es admitido en el grupo de procesos ejecutables del SO. Comúnmente, es un proceso que no ha sido cargado en la memoria principal a pesar de que su bloque de control de proceso (BCP) sí ha sido creado. Se corresponde con un proceso que acaba de ser definido. Mientras un proceso está en el estado Nuevo, la información relativa a éste que el SO requiera se mantiene en tablas de control de memoria principal. Pero el proceso en sí no está en la memoria principal.

Preparado: El proceso está listo para que lo asignen a un procesador. Está preparado para ejecutarse cuando tenga oportunidad.

En ejecución: Las instrucciones se están ejecutando. Asumimos que el equipo tiene un solo procesador, de modo que sólo un proceso pueda estar en este estado en un instante específico.

En espera: el proceso está a la escucha de que se dé un evento determinado, ya sea la terminación de una operación de E/S o la recepción de una señal.

Terminado: Termina la ejecución del proceso, ya que ha sido liberado del grupo de procesos ejecutables por el SO por haber sido detenido o abortado por alguna razón. Primero, el proceso termina al alcanzar su punto de finalización natural, cuando se aborta por un error no recuperable, o cuando otro proceso con autoridad apropiada provoca que el proceso se aborte. En este punto, el proceso no se puede elegir otra vez para su ejecución. Las tablas y otra información asociada con el trabajo se encuentran temporalmente preservadas por el SO, el cual proporciona tiempo para que programas auxiliares o de soporte tomen la información necesaria. Luego de esto, el SO no requiere mantener ningún dato del proceso y éste se borra del sistema.

Null -> Nuevo: Creación de un nuevo proceso para ejecutar un programa.

Admitido: El proceso es admitido para que el procesador lo ejecute. El SO está preparado para ejecutar un nuevo proceso. El SO fija un límite basado en la cantidad de procesos existentes o en la cantidad de memoria virtual que se podrá usar por dichos procesos existentes. Para que así no haya demasiados procesos activos y que se degrade el rendimiento del sistema.

Preparado-> En ejecución: El SO selecciona uno de los procesos que se encuentren en estado Preparado. Esta tarea la realiza el planificador.

Interrupción: Se produce una interrupción y otro proceso se ejecuta. La razón más común para esta transición es que el proceso en ejecución haya alcanzado el máximo tiempo posible de ejecución de forma ininterrumpida.

Salida: Se evalúan la ejecución del proceso y su finalización. La terminación mueve el proceso al estado Terminado.

En espera de suceso o de operación de E/S: Después de ejecutarse, el proceso espera que se dé un p

Fin de suceso o de operación de E/S: Fin de la ejecución del proceso sin necesidad de terminarla completamente, ya que pasará al estado de estar preparado para asignarse al procesador.

3- Descripción de los procesos y sus elementos:

Cada proceso se representa en el SO por medio de un bloque de control de proceso (BCP), el cual contiene muchos elementos de información asociados con un proceso específico y sirve como repositorio de cualquier información que pueda variar de un proceso a otro.

Si el procesador comienza a ejecutar un código de programa, cuya entidad en ejecución es un proceso, en cualquier instante del tiempo de ejecución, dicho proceso se caracteriza principalmente por los siguientes elementos:

- **Identificador:** para distinguirlo del resto de procesos.

- **Estado:** Si el proceso está corriendo, se encuentra en estado de *ejecución*. El estado puede ser: nuevo, preparado, en ejecución, en espera, detenido, etc.
- **Prioridad:** Nivel de prioridad respecto al resto de procesos.
- **Contador de programa:** Indica la dirección de la siguiente instrucción del programa a ejecutarse.
- **Registros de la CPU:** Varían en número y tipo dependiendo de la arquitectura de la máquina. En ellos se incluyen los acumuladores, registros de índice, punteros de pila, registros de propósito general y toda la información de los indicadores de estado. Dicha información de estado se guarda junto con el contador de programa para que, en caso de una interrupción, el proceso pueda seguir ejecutándose con normalidad.
- **Información de planificación de la CPU:** Incluye cualesquiera de los parámetros de planificación requeridos, entre los que se encuentran la prioridad del proceso y los punteros a las colas de planificación.
- **Información de gestión de memoria:** Contiene información sobre el valor de los registros base y límite, las tablas de páginas o de segmentos, todo en base al mecanismo de gestión de memoria empleado por el SO.
- **Punteros a memoria:** Incluye los apuntadores al código de programa y los datos que se asocian a dicho proceso, además de los bloques de memoria compartido con otros procesos.
- **Datos de contexto:** Consisten en datos que se hallan en los registros del procesador cuando el proceso se encuentra en ejecución.
- **Información de estado de E/S:** Son las peticiones de entrada y salida pendientes, dispositivos de E/S asignados a dicho proceso, un listado de los archivos que éste se encuentra usando, etc.
- **Información de auditoría:** En ella puede encontrarse la cantidad de tiempo de procesador y de reloj utilizados, así como límites de tiempo, registros contables, entre otros registros.

4. Control de procesos

El sistema operativo en la gestión de sus procesos debe conceder diferentes modos de ejecución, esta decisión es meramente para proteger el sistema operativo del uso de instrucciones en un contexto que no sea el adecuado.

Algunos procesadores incluyen modos de procesamiento de procesos, algunos de ellos son:

- Modo privilegiado: permite lectura y modificación de los registros, instrucciones E/S primitivas e instrucciones de gestión de memoria. Adicionalmente, los modos privilegiados también pueden acceder a bloques de memoria específicos/dedicados.
- Modo Usuario: los programas se suelen ejecutar en esta modalidad y es el que menos privilegios acostumbra tener.
- Modo sistema: también conocido como modo control o modo núcleo. Contiene las funciones más importantes del sistema, su funcionalidad se basa sobre todo en sistemas o aplicaciones que acceden a información sobre el funcionamiento del equipo o funcionamiento interno. Ver tabla 3.7.

Sobre los modos, el sistema debe ser capaz de detectar cuando se está ejecutando un proceso con unos u otros privilegios. Una instrucción además de indicar que es lo que debe hacer contiene información sobre sí mismo, su procedencia, permisos, etc.

Creación de procesos.

Cuando el sistema operativo decide generar un nuevo proceso ocurren una serie de eventos exclusivamente para ello:

1. **Asignación de un identificador:** se genera una entrada en la tabla de procesos que identifica al nuevo proceso
2. **Asignación del espacio:** Obtiene el espacio necesario para incluir los elementos de la imagen del proceso, se asigna espacio según el tipo de proceso o se pueden instanciar en su solicitud de creación. Si el proceso es generado por otro se le dice padre y al nuevo proceso hijo, el proceso padre tiene control sobre el proceso hijo, puede detenerlo consultarlo, eliminarlo y enviar datos mientras que el proceso hijo no puede realizar ninguna de estas acciones sobre el padre más que consultarlo.
3. **Inicialización del bloque de control:** Inicializa el estado del proceso del BCP, usualmente es asignado a 0 en todas sus entradas a excepción del contador que se obtiene como entrada del programa. Desde esta etapa se puede inicializar el estado del proceso desde “listo/suspendido” a cualquier otro. También se puede fijar la prioridad de ejecución, lo mismo aplica para recursos de E/S pero deben ser indicados explícitamente.

4. **Establecer enlaces:** Cuando el sistema operativo contiene cada cola del planificador como una lista enlazada, ordena al proceso en una cola especial para procesos “listos” para ejecución.

5. **Creación o expansión de estructuras de datos:** con propósito de auditoría se almacenan datos de los procesos para conocer su rendimiento.

Cambio de procesos.

La generación de procesos da pie al siguiente dilema “¿Cuándo es momento de alternar entre procesos?”.

El cambio de un proceso a otro en su ejecución puede ocurrir en cualquier momento durante el uso del sistema que pueden por el sistema o por algún otro proceso en ejecución. Existen 2 diferentes interrupciones “interrupciones” y “traps”.

Las “interrupciones” se dan cuando un evento externo independiente al proceso actual dispara un evento para la interrupción mientras que “traps” están enfocadas a errores de ejecución dentro del mismo proceso.

Cambio de estado del proceso.

Con la implementación de interrupciones es necesario poder restaurar la información de las mismas y así también asegurar que los tiempos de reloj y otros datos no sean modificados incorrectamente durante la transición de estados. En el cambio de estados no solo se ve involucrados los procesos, también se tiene que hacer un cambio en los datos de entorno con los que funciona el procesador, algunos de los eventos que ocurren en el cambio de estado son:

1. Salvar el estado del procesador (contador del programa y los registros que se usaban).
2. Actualizar el bloque de control del proceso que se está ejecutando a otro, este cambio también agrega datos para auditoría, así se puede conocer el motivo por el que el proceso fue interrumpido.
3. Mover el bloque de control a la cola de estados apropiada (según la interrupción hecha).
4. Selección de un nuevo proceso a ejecutar.
5. Actualizar el estado del proceso elegido a “ejecución”.
6. Actualizar estructuras de datos de gestión de memoria.
7. Cuando se va a retomar el estado suspendido, se deben restaurar todos los datos a como estaban al momento de la interrupción

Este proceso requiere más esfuerzos que los anteriores pues incluye el cambio del modo de ejecución del procesador.

Y ahora con todo esto.

¿Cuándo es la ejecución del sistema operativo?

Sabemos que el sistema operativo funciona como cualquier software, pues siguen siendo un conjunto de programas ejecutados y también que el sistema operativo da el foco del procesador a los procesos en ejecución, como tal depende del procesador para recuperar el control.

En sí mismo el sistema operativo tiene su propia región en memoria y su propia pila de procesos, también puede realizar todas las funciones que se necesiten como restaurar el contexto interrumpido.

Estas acciones son ejecutadas cuando se está en modo de usuario, porque el sistema operativo requiere de modos privilegiados para ejecutar su pila de procesos.

5- Definición de hilo.

Unidad de trabajo que incluye el contexto del procesador, es un flujo de ejecución de tareas que se ejecutan secuencialmente, también puede ser interrumpido para permitir paso a un hilo diferente en su ejecución. Se pueden generar diferentes hilos de ejecución para un solo programa a conveniencia del desarrollo del mismo.

6- Hilo vs. Proceso.

Si comparamos a los hilos y los procesos nos daremos cuenta que los procesos son una colección de hilos. Ambos conceptos generan al programa en ejecución.

Vemos al proceso como un contenedor que representa un programa en ejecución y al aplicar uno o más hilos tenemos mejor manejo en la modularidad de un sistema.

Si bien es posible ejecutar un sistema informático con 1 solo hilo, se tienen muchas limitaciones en rendimiento cuando no existen más hilos a disposición, el sistema se ve limitado a alternar recursos a las tareas que tenga pendientes y se deberá someter a esperas forzadas si no se tiene cuidado con la gestión, en cambio, cuando se hace la implementación multihilo se pueden realizar más tareas a la vez, lo que significa un aumento al desempeño del sistema, a la disponibilidad (en cuanto a fallos del hardware) y escalabilidad del sistema.

7- Gestión de hilos.

Existen diferentes estrategias para la gestión de hilos algunas se ejecutan a nivel de usuario (**ULT user-level threads**) y otras a nivel de kernel (**KLT kernel-level threads**).

Los procesos ejecutados a nivel de usuario son completamente gestionados en este nivel, no requieren el modo núcleo del procesador para esto, de hecho, el

procesador no se da cuenta de la ejecución de hilos cuando esta se hace a nivel de usuario.

Esta actividad se hace desde una biblioteca proporcionada por el mismo sistema que permite generar estructuras de datos para generar hilos, esta clase de bibliotecas permiten gestionar hilos como lo haría el sistema en sí, pueden crear y eliminar hilos, enviar datos entre hilos, planificar la ejecución y guardar y restaurar el estado de los hilos (en caso de interrupciones) estas algunas funcionalidades son declaradas explícitamente por el desarrollador que aprovecha la librería pero otras son realizadas por la misma biblioteca.

La gestión hilos a través del espacio de usuario se realiza dentro del contexto de un proceso en particular. y la biblioteca se encarga de realizar el trabajo con el sistema operativo en cuanto al cambio de contexto, prioridad de otros hilos y reanudar la ejecución del proceso mismo. También es posible que un hilo se encuentre en estado de ejecución pero el proceso en sí esté bloqueado, listo para ejecutar o en ejecución, principalmente esta aparente descoordinación entre estados se debe a la transición que se realiza entre ejecución de un proceso a otro como se ve en la siguiente imagen.

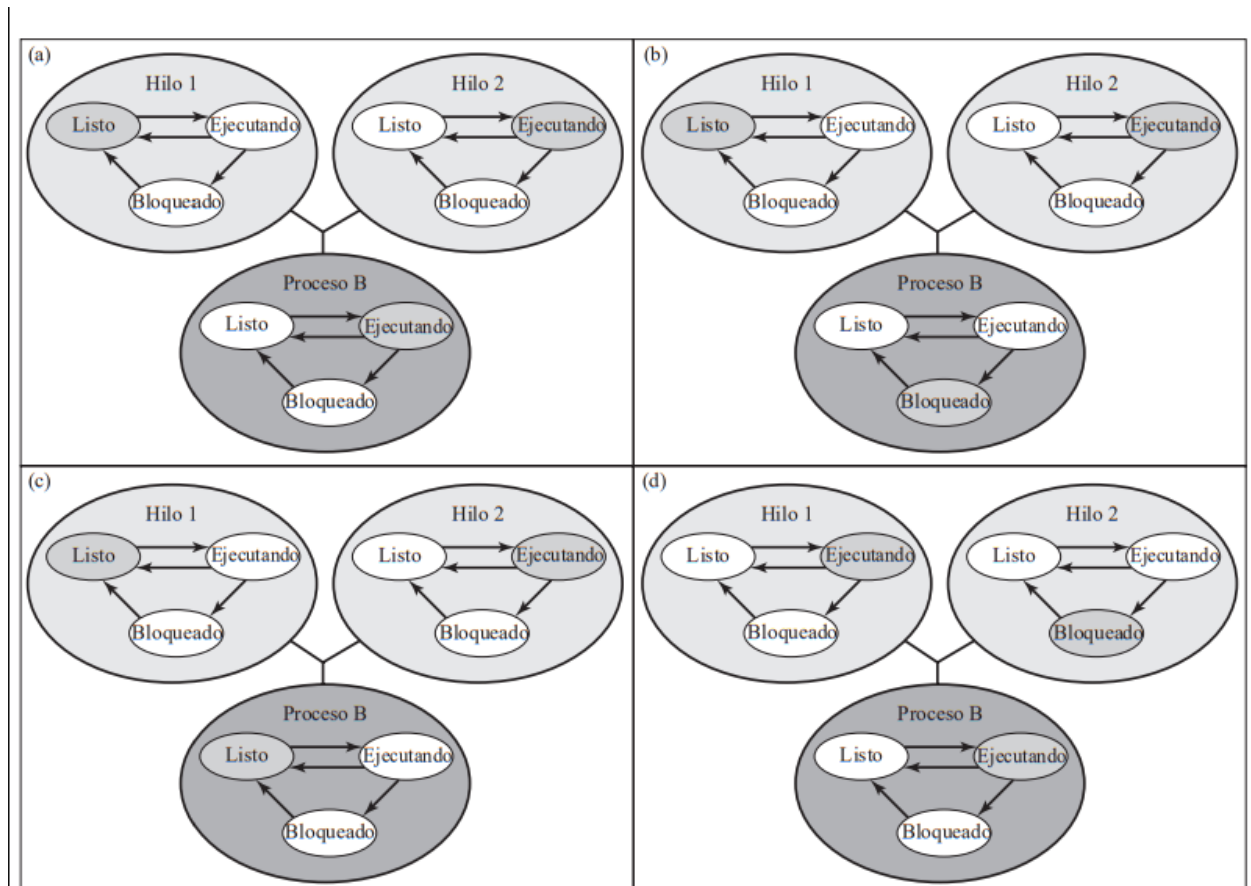


Ilustración 2.- Transición entre ejecución de un proceso a otro.

Esta estrategia toma fuerza en los siguientes aspectos:

1. No requieren privilegios para cambiar de hilo en el procesador, por que todas las estructuras ya están en un espacio asignado a modo usuario. Permite ahorrar esfuerzo en el cambio de modalidades.
2. La planificación se realiza desde el software en ejecución y no requiere modificar al sistema operativo.
3. Al ser bibliotecas de utilidades de aplicación se pueden ejecutar en cualquier sistema operativo.

Si bien sus ventajas son bastante aprovechables para desarrollos de terceros, no debemos olvidar que existen desventajas al respecto como:

1. En sistemas operativos típicos, muchas llamadas al sistema son bloqueantes, dando pie a que no solo se bloquee un único hilo, sino que se bloquean todos los hilos de proceso.
2. No se puede sacar provecho al multiproceso, pues el proceso se ejecuta en un solo núcleo del procesador y el multihilo sucede dentro de esa misma asignación, no resulta un problema para todas las aplicaciones existentes pero en ocasiones en que se requiera la ejecución de un código concurrente, se puede ver afectado el desempeño.

Durante la gestión multihilo KLT el núcleo de procesador realiza el trabajo, no existe código programado para la gestión en la aplicación, solo una interfaz de programación.

De entrada el núcleo mantiene información del contexto del proceso, significando que se deben realizar cambios en la modalidad de operación del procesador que en sí mismo ya es una gran desventaja pero tiene a favor que su planificación es a nivel de hilo individual por proceso, que es una gran ventaja que no hace independiente la ejecución del hilo sobre el estado del proceso.

Ambas estrategias tienen limitaciones muy notorias pero ¿qué pasaría si tomáramos lo mejor de cada una y combinamos su operación?

En un **enfoque combinado** se pueden generar los hilos a nivel de usuario igualando los en número de KLT, así se consigue seguir ejecutando hilos de un mismo proceso en paralelo en diferentes procesadores sin que el estado del proceso limite su procesamiento.

Conclusiones

Estrada Ceja Ulises Iván:

La necesidad de aislar la ejecución de los programas debido a la posibilidad de cargar en memoria múltiples tareas y ejecutarlas simultáneamente dio lugar a los procesos, que se componen de distintos elementos y se controlan mediante mecanismos como la planificación, creación, finalización y comunicación. Los procesos se caracterizan por tener un contador de programa que indica la siguiente instrucción que se debe ejecutar y por tener a la mano ciertos recursos. Los procesos pueden pasar por distintos estados mientras se ejecutan, dichos estados se determinan por su actividad actual y sólo puede haber un proceso ejecutándose en un CPU en un instante concreto, pero puede haber muchos procesos en espera. El SO visualiza los procesos con un bloque de control de procesos que contiene varios elementos de información de un proceso específico.

Hernández Lomelí Diego Armando:

Un sistema operativo ejecuta una cantidad masiva de procesos durante su uso y para tal cantidad es necesario optar por una estrategia de ejecución y no solo eso, las mismas características que surgen con nuevos desarrollos hacen que sea necesario emplear más recursos del procesador para una única tarea. Hoy en día estas bases están más que establecidas en los sistemas operativos y en el hardware que se emplea, pero es de mucha ayuda sobre todo a futuros desarrolladores conocer la estructura en la que se concibe un proceso y las estrategias en las que se distribuye la ejecución de procesos sobre todo para evitar realizar esfuerzos adicionales no necesarios y optimizar los recursos disponibles.

Bibliografía

Silberschatz, A. Galvin, P., B.Gagne, G. (2006). Procesos. En C. Sánchez González (Ed.), *Fundamentos de sistemas operativos. Séptima edición*. (pg. 73 - 111). Mc Graw-Hill / Interamericana de España, S. A. U.

Stallings, W. (2005). Descripción y control de procesos. En M. Martín-Romo (Ed.), *Sistemas operativos. Aspectos internos y principios de diseño. Quinta edición*. (pg. 107 - 150). Pearson Educación, S. A.

Anexos:

Sin anexos.